CS 830: Randomness and Complexity 03/01/05

Lecture 13: Nisan's Amplification

instructor: Jin-Yi Cai Scribe: Scott Diehl, Jeff Kinne

Today we complete the presentation of Nisan's randomness-efficient amplification of an RP algorithm, and begin to discuss ideas and definitions that will lead to the presentation of Impagliazzo and Zuckermann's leftover hash lemma.

1 Nisan's Amplification

We start by restating a technical lemma proved last time, which says that for any set A, the probability that $(y, h_s(y)) \in A \times A$ is close to the probability that $(y, z) \in A \times A$ for y, z uniformly chosen, for all but a very few universal hash functions h_s.

Lemma 1.1 (Hash Mixing Lemma). *Let $\{h_s\}$ be a family of universal hash functions specified by S, $h_s : \{0,1\}^n \to \{0,1\}^n$, and*

$$\epsilon = 2^{-n/3}. \tag{1}$$

For all sets $E \subseteq (\{0,1\}^n)^2$, for all but a fraction of $\epsilon/4$ many $s \in S$,

$$|\Pr[y \circ h_s(y) \in E] - \mu(E)| < \epsilon. \tag{2}$$

We call the at most $\epsilon/4$ fraction of the h_s for this which does not hold "bad", and the rest "good".

Recall that the generator we are using for this amplification is constructed recursively as follows, given $s_1, s_2, \ldots, s_i$:

$$\begin{aligned}
G_0(y) &= y \\
G_1(y) &= y \circ h_{s_1}(y) \\
G_2(y) &= y \circ h_{s_1}(y) \circ h_{s_2}(y) \circ h_{s_1}(h_{s_2}(y)) \\
&\cdots \\
G_i(y) &= G_{i-1}(y, s_1, \ldots, s_{i-1}) \circ G_{i-1}(h_{s_i}(y), s_1, \ldots, s_{i-1}).
\end{aligned}$$

We can see that the generator G_k, given k universal hash functions, stretches the n bit input y into $n \cdot 2^k$ bits. Since we can specify universal hash functions on n bits by $2n$ bits, G_k produces 2^k trials for an RP algorithm from $(1 + 2k)n$ input bits. In other words, it requires $O(n \log \ell)$ random bits to generate ℓ trials.

How close to independent trials do those generated by G_ℓ act? We show that up to a small error term, they are just as good, which implies that we can obtain an exponentially small error probability for our RP algorithm using only $O(n \log n)$ random bits. The proof goes by induction on the "level" k of the generator G_k, where at each step we use Lemma 1.1 to bound the error that the hash function h_{s_k} is bad, as well as the probability that a "good" hash function causes both y and $h_{s_k}(y)$ to both land in a set of bad witnesses defined by the previous $s_1, \ldots, s_{k-1}$.

THEOREM 1.2 (NISAN'S GENERATOR). *Let $L \in \mathrm{RP}$. For $x \in L$, let W_x be the set of witnesses of x for an* RP *machine recognizing L. Then*

$$\Pr[G_k(y, s_1, \ldots, s_k) \subseteq \overline{W_x}] \leq \mu(\overline{W_x})^{2^k} + (k/4+2)\epsilon, \tag{3}$$

where ϵ is defined in (1).

Proof. For fixed $s_1, \ldots, s_{i-1}$, the set of bad witnesses with respect to these choices is

$$A_i = \{y | G_{i-1}(y, s_1, \ldots, s_{i-1}) \subseteq \overline{W_x}\}.$$

For each i, we want to consider the probability that $y \circ h_{s_i}(y)$ lands in $A_i \times A_i$. We will refer to s_i as "good" when (2) holds for $E = A_i \times A_i$, and "bad" when this does not hold. When s_i is "bad", we really can't say anything about the distribution of $y \circ h_{s_i}(y)$ compared to $\mu(A_i \times A_i)$, so we would like to consider only "good" s_i. Therefore, we can use the union bound to bound the probability that s_1 is bad, s_2 is bad given that s_1 is good, s_3 is bad given that s_1 and s_2 are good, and so on. By Lemma (1.1), the probability that each s_j is bad is at most $\epsilon/4$. Therefore,

$$\Pr[\text{any } s_1, \ldots, s_k \text{ is bad}] \leq k\epsilon/4. \tag{4}$$

In order to finish our computation, we must bound the probability that $y \circ h_{s_i}(y)$ lands in $A_i \times A_i$ given that $s_1, \ldots, s_i$ are all good. We claim that

CLAIM 1.3. *When $s_1, \ldots, s_i$ are good (in terms of Lemma 1.1),*

$$\mu(A_{i+1}) = \Pr_y[G_i(y, s_1, \ldots, s_i) \subseteq \overline{W_x}] \leq \beta^{2^i} + 2\epsilon, \tag{5}$$

where $\beta = \mu(\overline{W_x})$.

Before proving the claim, note that setting $i = k$, we can bound the probability that $G_k(y, s_1, \ldots, s_k) \subseteq \overline{W_x}$ by adding the failure probability (5) given that $s_1, \ldots s_k$ are good, and the probability (4) that any s_j is bad, which results in exactly (3). Therefore, all that remains to complete the proof is to prove the claim.

We prove the claim by induction on i. For $i = 0$, the bound holds trivially, since $G_0(y) = y$. For $i = 1$, $G_1(y) = y \circ h_{s_1}(y)$, so by applying Lemma 1.1 with $E = A_1 \times A_1 = \overline{W_x} \times \overline{W_x}$, we have error probability $\epsilon + \mu(E) = \epsilon + \beta^2$. We now argue the inductive case $i - 1 \to i$. By Lemma 1.1, since s_i is good, the error probability $\mu(A_{i+1})$ is bounded by $\epsilon + \mu(A_i \times A_i)$, where A_i is defined by the good $s_1, \ldots, s_{i-1}$. By the inductive hypothesis, $\mu(A_i) \leq \beta^{2^{i-1}} + 2\epsilon$, so that $\mu(A_i \times A_i) \leq (\beta^{2^{i-1}} + 2\epsilon)^2$. By our choice of ϵ, we can see that

$$\mu(A_{i+1}) \leq \epsilon + (\beta^{2^{i-1}} + 2\epsilon)^2 \leq \beta^{2^i} + 2\epsilon,$$

as desired. ∎

2 Definitions

We now give some definitions and results that will be needed in the next few lectures.

DEFINITION 2.1. *Given a distribution P on a set S, we use the following as shorthand notation:*

$$p_x = \Pr_{Y \in_P S}[Y = x].$$

In other words p_x is the probability that x occurs when sampling according to P.

DEFINITION 2.2 (INFORMATION). *Given a distribution P on a set S, then the information of x according to P is:*

$$\forall x \in S, inf_P(x) = -\log(p_x).$$

If $p_x = \frac{1}{2^a}$, then $inf_P(x) = a$. $inf_P(x)$ can be thought of the number of bits of information needed to describe x's probability in P.

DEFINITION 2.3 (ENTROPY). *Let X be a random variable drawn according to a distribution P over a set S. Then the entropy of X is:*

$$H(X) = -\sum_{x \in S} p_x \log(p_x).$$

DEFINITION 2.4 (RENYI ENTROPY). *Let X be a random variable drawn according to a distribution P over a set S. Then the the Renyi entropy of X is:*

$$H_{ren}(X) = -\log(\sum_{x \in S} p_x^2).$$

Equivalently, $H_{ren}(X)$ corresponds to the probability of two independent random variables colliding ($\Pr[X = X']$ where X and X' are independent identically distributed random variables).

DEFINITION 2.5 (MIN-ENTROPY). *Let X be a random variable drawn according to a distribution P over a set S. Then the min-entropy of X is:*

$$H_\infty(X) = \min_{x \in S}\{-\log(p_x)\}.$$

This corresponds to the maximum p_x.

DEFINITION 2.6 (INFORMATION DIVERGENCE). *Let Y and X be random variables over a set S, with the probabilities of Y given by q and the probabilities of X given by p. The information divergence of Y with respect to X is:*

$$\sum_{i \in S}(p_i(-\log(q_i) + \log(p_i))).$$

Notice that: $\sum_{i \in S}(p_i(-\log(q_i) + \log(p_i))) = \sum_{i \in S}(p_i(-\log(\frac{q_i}{p_i}))) \geq \sum_{i \in S}(p_i(1 - \frac{q_i}{p_i})) = 0$. In other words, the information divergence is always non-negative.

EXERCISE 2.7. *Show that for any random variable X on $S = \{0,1\}^n$, $H(X) \leq n$. Hint: use the fact that $\forall z > 0, \log_e z \leq (z-1)$.*

EXERCISE 2.8. *Show that if random variables X, Y are independent, then $H(X \cdot Y) = H(X) + H(Y)$.*

EXERCISE 2.9. *Show that for all random variables X, Y, $H(X \cdot Y) \leq H(X) + H(Y)$.*

Exercise 2.10. *Show that for all random variables X, $H_{ren}(X)/2 \leq H_\infty(X) \leq H_{ren}(X) \leq H(X)$.*

Definition 2.11 (Statistical distance). *Given 2 distributions P, Q on the same set S, with the probabilities according to P denoted p and the probabilities according to Q denoted q, the statistical distance of P and Q is:*

$$\max_{T \subseteq S}\{\sum_{i \in T}(p_i - q_i)\}.$$

Notice that (inventing notation), $\max_{T\subseteq S}\{\sum_{i\in T}(p_i - q_i)\} = \max_{T\subseteq S}\{P(T) - Q(T)\} = \max_{T'\subseteq S}\{Q(T') - P(T')\}$. That is, the definition of statistical difference is symmetric (which is why we did not need to use absolute values in the definition).

Definition 2.12 ($\ell - k$ norm). *For $k > 0$, we define the $\ell - k$ norm:*

$$||p - q||_k = \sqrt[k]{\sum_i |p_i - q_i|^k}.$$

Notice that the statistical distance between distriutions P and Q is equal to $\frac{||p-q||_1}{2}$.

Theorem 2.13 (Cauchy-Schwarz). *If u and v are members of an inner product space, then*

$$|\langle u, v\rangle|^2 \leq \langle u, u\rangle \cdot \langle v, v\rangle,$$

or equivalently

$$|\langle u, v\rangle| \leq ||u||_2 \cdot ||v|_2.$$

Definition 2.14 (ϵ-uniform). *A distribution P on a set S, is ϵ-uniform if $\sum_{i\in S} |p_i - \frac{1}{|S|}| \leq \epsilon$. Equivalently, each point's probability is at most $\frac{\epsilon}{2}$ from uniform.*

3 Impagliazzo-Zuckerman Generator and Leftover Hash Lemma

Given an RP language, we would like to generate a sequence of random bits to look for a witness for the language. We wish to have a high probability of finding a witness while using the fewest number of random bits. The Impagliazzo-Zuckerman generator is one method for generating witnesses. The idea is to begin with a uniform set of random bits. At each further stage, we will reuse bits from the previous stage. The idea is that at each stage, not all of the random bits are used - there are some that are "leftover" for future stages. We will generate random strings Y as follows:

Choose $Y_1 \in \{0,1\}^n$ and $h_s : 2^n \to 2^{n-l}$ at random.
Choose $z_i \in \{0,1\}^l$ at random, and set $Y_i = h_s(Y_{i-1}) \circ z_i$.

At each stage other than the first, we will be using $l < n$ random bits and reusing random bits from the previous stage by hashing. To prove the effectiveness of this generator, we will need some lemmas about hash functions.

Lemma 3.1. *For a distribution P on a set S, if the collision probability $c(P) \leq \frac{1+\epsilon^2}{|S|}$, then P is ϵ-uniform.*

Proof. We will show that $(\sum_{i\in S}|p_i = \frac{1}{|S|}|)^2 \leq \epsilon^2$ which is equivalent to showing that P is ϵ-uniform.

$$\begin{aligned}
(\textstyle\sum_{i\in S}|p_i - \frac{1}{|S|}|)^2 &\leq (\textstyle\sum_i (p_i - \frac{1}{|S|})^2 (\sum_i 1^2)) \\
&= (\textstyle\sum_i (p_i^2 - \frac{2p_i}{|S|} + \frac{1}{|S|^2}))|S| \\
&= ((\textstyle\sum_i p_i^2) - \frac{2}{|S|} + \frac{1}{|S|})|S| \\
&= (c(P) - \tfrac{1}{|S|})|S| \\
&\leq \tfrac{\epsilon^2}{|S|}|S| = \epsilon^2.
\end{aligned}$$

The first inequality is due to the Cauchy-Schwarz inequality. □

The following lemma intuitively describes the amount of "leftover" randomness from a uniform sample that can be reused by hashing.

Lemma 3.2 (Leftover Hash Lemma). *Let H be a family of universal hash functions, with $h_s(y): Y \to T$, $s \in S$, and $Y' \subseteq Y$. Let P be the distribution $s \circ h_s(y)$ where s is taken uniformly from S and y is taken uniformly from Y'. Then*

$$c(P) \leq \frac{1 + |T|/|Y'|}{|S \times T|}.$$

Applying the previous lemma, this means that P is ϵ-uniform where $\epsilon = \sqrt{|T|/|Y'|}$.

We will finish this analysis and give the result about the Impagliazzo-Zuckerman generator in the next lecture.